



Universidade Federal de Goiás
Instituto de Informática
Sistemas de Informação

Matriz Curricular: SIINF-BN-2 - 2017.1

Plano de Disciplina
Ano Letivo: 2022 - 1º Semestre

Dados da Disciplina

Código	Nome	Carga Horária	
		Teórica	Prática
10000083	Programação Orientada a Objetos - NBC	32	32

Prof(a): Nadia Felix Felipe da Silva

Turma: A

Ementa

Abstração e tipos abstratos de dados. Classes, métodos, encapsulamento, interface. Mensagens, instâncias e inicialização. Herança e composição. Polimorfismo. Uso de uma linguagem orientada a objetos. Noções de UML. Noções de padrões de projeto orientado a objetos.

Objetivo Geral

A disciplina deve prover ao estudante condições de compreender o paradigma de programação orientada a objetos, assim como contextualizar e aplicar tal paradigma na elaboração de programas computacionais.

Objetivos Específicos

De forma específica, a disciplina deve possibilitar aos estudantes a capacidade de: Distinguir o modelo de orientação a objetos do modelo imperativo / estruturado; Explicar os principais conceitos e fundamentos da programação orientada a objetos; Resolver problemas utilizando o conceito da orientação a objetos; Capacitar o acadêmico em relação a análise de problemas reais, abstração e modelagem de programas computacionais usando os conceitos de orientação a objetos; Implementar programas orientados a objeto com um ambiente de desenvolvimento integrado; Usando os recursos como teste de unidade e controle de versão; Implementar programas utilizando interface gráfica e streams; Elaborar modelos utilizando UML; Compreender as visões da arquitetura de software.

Relação com Outras Disciplinas

O bom aproveitamento da disciplina Programação Orientada a Objetos depende do conhecimento das disciplinas Introdução a Programação e Algoritmos e Estrutura de Dados 1. Esta disciplina possui uma relação de continuidade com as disciplinas citadas, uma vez que promove o aperfeiçoamento dos alunos em termos de programação, além de trabalhar um modelo de programação diferente daquele abordado nas duas disciplinas anteriores.

A visão e o conhecimento do modelo de objetos obtidos pelo estudante na disciplina Programação Orientada a Objetos é essencial a outras disciplinas de programação oferecidas no decorrer do curso como Algoritmos e Estruturas de Dados 2, Projeto de Software, Linguagens e Paradigmas de Programação e Desenvolvimento de Software para Web.

Programa

I - Programação orientada a objetos: modelo de objetos, tipos abstratos de dados, classes, métodos, visibilidade, variáveis de instância, métodos, mensagens, construtores, escopo, associação entre classes, composição e delegação, coleções, herança de classes, interfaces, classes abstratas, polimorfismo.

II - Tópicos: tratamento de exceções (Categorias, tratamento, captura e lançamento); uso de streams (tratamento de arquivo); construção de aplicações usando a Linguagem Java com testes de unidade (JUnit), depuração e controle de versão (Git); interfaces gráficas.

III - Noções de Projeto: Introdução a UML (Diagrama de Classes); padrões de projeto (GoF – Gang of

Four);

Procedimentos Didáticos

Legenda	Descrição	Objetivo
AEX	Aula teórica	Transmitir conhecimento utilizando quadro ou slides.
AP	Aula prática	Proporcionar ao aluno a aplicação prática do conteúdo ministrado em aula teórica.
ED	Estudo dirigido	Desenvolver a capacidade analítica, capacidade de síntese, de avaliação crítica e de análise.
OTR	Outros	Transmitir conhecimento utilizando quadro ou slides.
RE	Aula teórica com resolução de exercícios	Desenvolver o raciocínio lógico, criatividade e capacidade de abstração e a capacidade de identificar, analisar e projetar soluções de problemas.
SE	Seminários	Desenvolver o raciocínio lógico, criatividade, capacidade de abstração, capacidade para identificar, analisar, projetar soluções de problemas, a capacidade de comunicação oral e a capacidade de trabalhar em grupo.
TG	Trabalho em grupo	Desenvolver a capacidade de comunicação oral e escrita. Capacidade de trabalhar em grupo.

Conteúdo Programático / Cronograma

Início	Proc. Didático	Tópico	# Aul.
30/05/22	AEX, AP, ED	Apresentação da Disciplina, História da Orientação a Objetos, Fundamentos de POO. Laboratório com Exercícios mostrando a diferença entre os paradigmas. Conceitos de POO: Classes, Objetos, Atributos e Métodos, Construtores, Estado de um objeto, interação entre objetos. Representação em UML. Exercícios de criação de classes e instâncias. Atividade Supervisionada : exercícios de implementação a ser entregue pelo SIGAA.	4
06/06/22	AEX, AP, ED	Conceitos de POO: continuação do conteúdo da aula anterior, encapsulamento, modificadores de acesso, métodos de acesso, atributos e métodos estáticos. Representação em UML. Exercícios usando os conceitos da aula teórica. Atividade Supervisionada: exercício de implementação a ser entregue pelo SIGAA.	4
13/06/22	AEX, AP, ED	Conceitos de POO: Associação entre objetos, herança, composição, agregação. Representação em UML. Exercícios de laboratório. Atividade Supervisionada: exercício de implementação a ser entregue pelo SIGAA.	4
20/06/22	AEX, AP, ED	Introdução a UML e principais componentes de um Diagrama de Classes. Exercícios de laboratório. Atividade Supervisionada: exercício de implementação a ser entregue pelo SIGAA.	4
27/06/22	RE, AP, ED	Projeto (1ª parte) - aula dedicada ao desenvolvimento da primeira parte do projeto. Atividade Supervisionada: entregue pelo SIGAA	4
04/07/22	OTR	1ª Prova Individual.	4
11/07/22	AEX, AP, ED	Conceitos de POO: Classes Abstratas, Interfaces, Polimorfismo. Representação em UML. Exercícios de laboratório. Atividade Supervisionada: exercício de implementação a ser entregue pelo SIGAA.	4
18/07/22	AEX, AP, ED	Introdução a Padrões de Projeto: Categorias, exemplos de implementações. Exercícios de laboratório. Atividade Supervisionada: exercício a ser entregue pelo SIGAA.	4
25/07/22	AEX, AP, ED	Controle de Exceções, Tratamento de Arquivos e Persistência. Exercícios de laboratório. Atividade Supervisionada: exercício a ser entregue pelo SIGAA.	4
04/08/22	AEX, AP, ED	Introdução aos testes de software. Testes Unitários. Exercícios de laboratório. Atividade Supervisionada : exercício a ser entregue pelo SIGAA.	4
08/08/22	OTR	2ª Prova Individual.	4
15/08/22	AEX, AP, ED	Projeto (2ª Parte): aula dedicada à 2ª parte do projeto. Tópicos Avançados para o Projeto Final. Atividade Supervisionada : a ser entregue pelo SIGAA.	4
22/08/22	TG, ED, AP	Realização SALT - Ainda em data a definir	4



Início	Proc. Didático	Tópico	# Aul.
29/08/22	TG, ED, AP	Apresentação de Trabalho Final. Prova Substitutiva. Atividade Supervisionada : esclarecimento de dúvidas pelo fórum da disciplina.	8
12/09/22	TG, ED, AP	Vista de prova	4
Total			64

Critério de Avaliação

Componentes que integram a avaliação da disciplina:

(APS_LTI) Atividades Práticas Supervisionadas (laboratórios, listas de exercícios e trabalhos individuais) relacionadas à resolução de problemas.

(APS_TG) Atividade Prática Supervisionada (Trabalhos em Grupo) para o Projeto de Aplicação.

(P1) 1a. Prova Individual.

(P2) 2a. Prova Individual.

(PS) Prova Substitutiva para quem faltar uma das duas provas individuais com justificativa.

A média final MF será assim calculada: $MF = (APS_LTI \cdot 0.1) + (APS_TG \cdot 0.2) + (P1 \cdot 0.35) + (P2 \cdot 0.35)$

- Avaliação do SALT - Aplicada exclusivamente aos estudantes do curso de Bacharelado em Sistemas de Informação.

- Aplicação do SALT 2022/1 (avaliação interdisciplinar) - Em agosto/2022 (ainda a definir dia), das 19h às 22h, com a participação de todas as turmas do curso.

- A prova será impressa, aplicada em sala de aula.

Observações:

(1) As notas das Atividades Práticas Supervisionadas e das Provas terão valor de zero a dez.

(2) As Atividades Supervisionadas referem-se a atividades práticas e devem ser aplicadas segundo Resolução CNE/CES 03/2007 de 2 de julho de 2007, a qual considera que os Bacharelados do período noturno dividem cada hora de atividade acadêmica em 45 minutos de preleções e aulas expositivas e 15 minutos de atividades práticas supervisionadas que podem ser realizadas a distância ou não, mas com supervisão do professor.

No contexto desta disciplina, elas consistem nas listas de exercícios a serem feitas no período entre as aulas presenciais e no desenvolvimento de um Projeto de Aplicação durante todo o semestre.

(3) O Projeto de Aplicação trata-se de uma aplicação completa a ser definida pelo professor, que deverá ser desenvolvida individualmente ou em duplas.

(4) Todas as atividades serão recebidas apenas pela plataforma Turing.

(5) As Provas P1, P2 serão individuais e cobrirão o conteúdo desenvolvido até a data de sua aplicação. Será atribuída a nota 0,0 (zero) a qualquer atividade ou trabalho não realizado ou não entregue na data estipulada. Será atribuída a nota 0,0 (zero) a qualquer atividade ou trabalho que for considerada(o) cópia, durante a correção. O pedido de segunda chamada deverá seguir o estipulado pelo RGCG. O aluno que não comparecer a pelo menos 75% das aulas estará reprovado por falta. O aluno que não conseguir média final maior ou igual a 6,0 (seis) estará reprovado por média.

Data da Realização das Provas

1a. Prova (P1): 04/07/2022. SALT: Agosto: em data a ser definida pelo NDE do curso (19h às 22h). 2a. Prova (P2): 22/08/2022. Prova Substitutiva (PS): 05/09/2022.

Local de Divulgação dos Resultados das Avaliações

Sala de Aula e SIGAA.

Bibliografia Básica

ECKEL, B. Thinking in Java. Prentice Hall 3rd ed . 2002. (livro eletrônico de acesso livre : disponível em <http://www.mindview.net/Books/TIJ/>).

BORATTI, I. C. Programação orientada a objetos em Java. Visual Books, 2007.

DEITEL, P. J; DEITEL, H. M. Java como programar 6.ed. São Paulo:Prentice Hall, 2005.

Bibliografia Complementar

ZEIGLER, B. P. Objects and Systems: Principled Design with Implementations in C++ and Java. New York, NY, USA: Springer-Verlag, Inc., 1997.

SANTOS, R. Introdução à Programação Orientada a Objetos com Java. Campus, 2003.

BUDD, T. An Introduction to Object-Oriented Programming. Addison Wesley, 1996.

HORSTMANN, C. S. Core Java – Advanced Features. 8. ed. Prentice Hall, 2008.(Volume II).
GAMMA, E. Design patterns elements of reusable object-oriented software. Reading:Addison Wesley, 1995.

Bibliografia Sugerida

DEITEL, H. M., Java Como Programar, Pearson Prentice Hall Brasil, 8a. edição, 2010.

Termo de Entrega	Termo de Aprovação
Apresentado à Coordenação no dia	Aprovado em Reunião de CD no dia
Prof(a) Nadia Felix Felipe da Silva Professor	Prof. Dr. Eliomar Araújo de Lima Diretor do Instituto de Informática
Termo de Homologação	
Data de Expedição: Goiânia, ____ de _____ de ____.	

